IBM Capstone Project

September 2020

Bibhash Chandra Mitra

Introduction 2

Data 2

Methodology 3

- Data Pre-processing 3
- Selecting features 3
- Adding new features 3
- Handling missing values 3
- One Hot Encoding 4
- Sampling the Data 4

Data Analysis and Visualization 5

- Seattle Accident Map 5
- Visualization based on various Columns 6

Deployment Modelling and Evaluation 9

- K-Nearest Neighbors (KNN) 9
- Decision Tree 10
- Random Forest 10

Results 11

Discussion 11

Conclusion 12

Introduction

There are many factors which play a role in any type of accident. We cannot pinpoint a single reason behind the severity of an accident. But out of all those factors there are some which can be tracked and are definite in nature. For example, the weather condition, light condition, type of sidewalk etc are some of the factors which are deterministic. Other factors like number of pedestrians, speeding etc are not deterministic without a high speed camera so we won't mention them. So can we say that using these factors we can answer questions like: **What is the probability of an accident occurring on a rainy day at an Intersection?**

I believe we can answer these types of questions and hence the purpose of this project is to devise a solution for the prediction of severe car accidents based on the different conditions which can be effectively tracked.

Data

For this project I have used the **Seattle transit Collision dataset**

This dataset contains all the types of collisions from 2004 to present. There are 38 columns in this dataset, out of those we will use the following:

- JUNCTIONTYPE : Category of junction at which collision took place
- WEATHER : A description of the weather conditions during the time of the collision.
- ROADCOND : The condition of the road during the collision
- LIGHTCOND : The light conditions during the collision.
- INCDTTM : The date and time (From which we can get information on the month and time of the day)
- X : Longitude of the location
- Y : Latitude of the location
- ADDRTYPE : Collision address type:•Alley •Block •Intersection

- SEVERITYCODE : A code that corresponds to the severity of the collision: •3—fatality •2b—serious injury •2—injury •1—prop damage •0—unknown

Methodology

Data Pre-processing

Main objective of this step is to get the pre-selected variable for machine learning. It includes the steps Exploratory Data Analysis, dealing with missing values, selecting features and converting the data types.

Selecting features

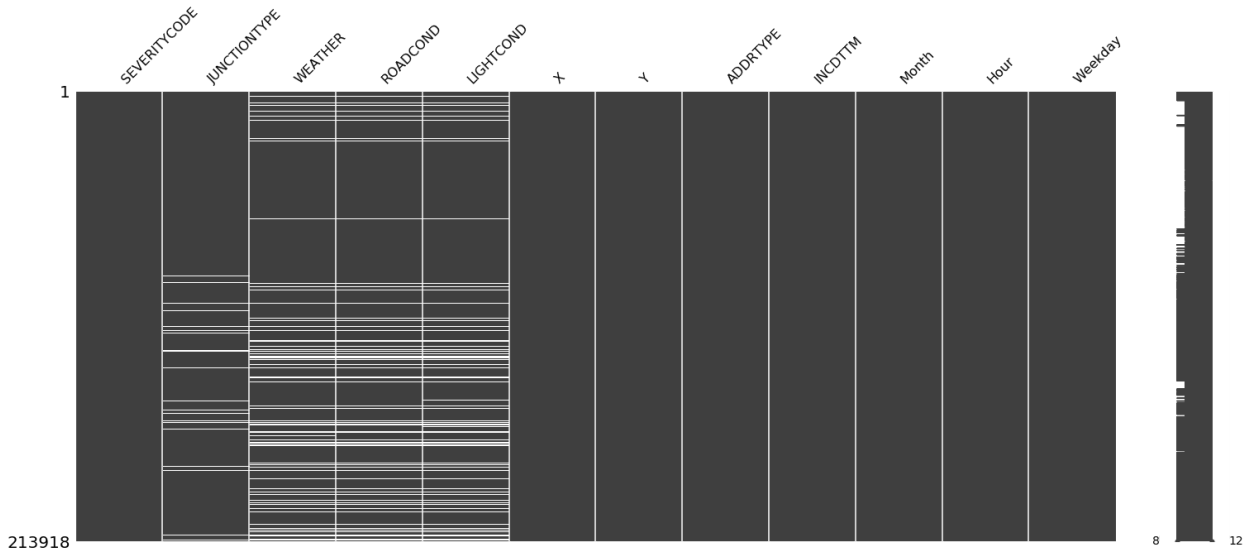
For features we have used the following columns as they are the only descriptive features and can be tracked:

- JUNCTIONTYPE : Category of junction at which collision took place
- WEATHER : A description of the weather conditions during the time of the collision.
- ROADCOND : The condition of the road during the collision
- LIGHTCOND : The light conditions during the collision.
- INCDTTM : The date and time (From which we can get information on the month and time of the day)
- ADDRTYPE : Collision address type:•Alley •Block •Intersection

Adding new features

Converting **INCDTTM** which is a timestamp to the **month**, **hour** and **weekday** column can help us group many time dependent conditions present in the dataset.

Handling missing values



Missing values visualization

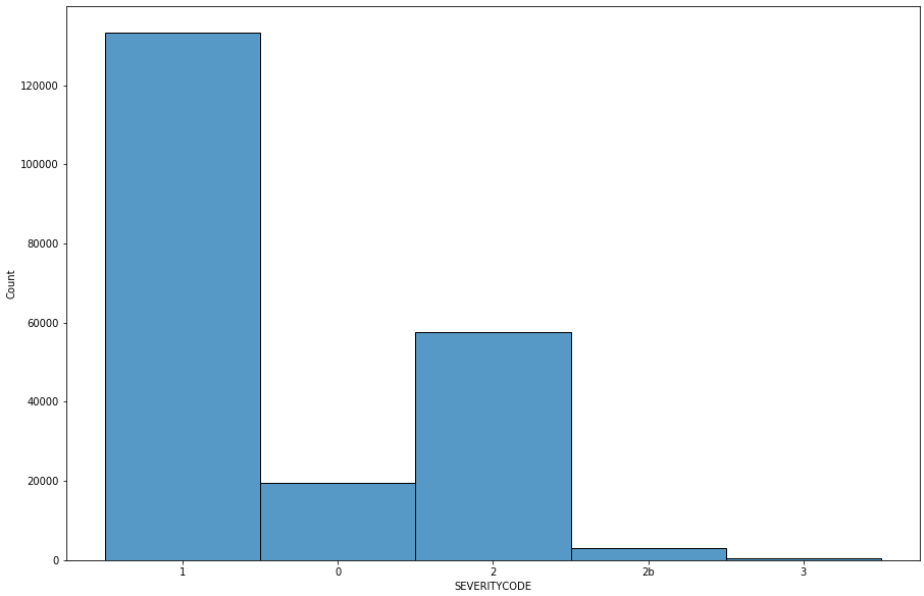
Filling the categorical rows which have missing values with "**UNKNOWN**" so that we can treat it as another category.

One Hot Encoding

As we know we cannot use the texts directly in any model, so we need to convert them into numbers which can be done using one hot encoding.

Sampling the Data

Through the EDA above, we can clearly notice that the class distribution in this dataset is very imbalanced. This is due to the fact that the lowest and highest severity accidents don't occur as often as compared to other two severities, so we don't have adequate data for those classes. This means if we used the data in its existing condition then the model may never give predictions which have those probabilities.



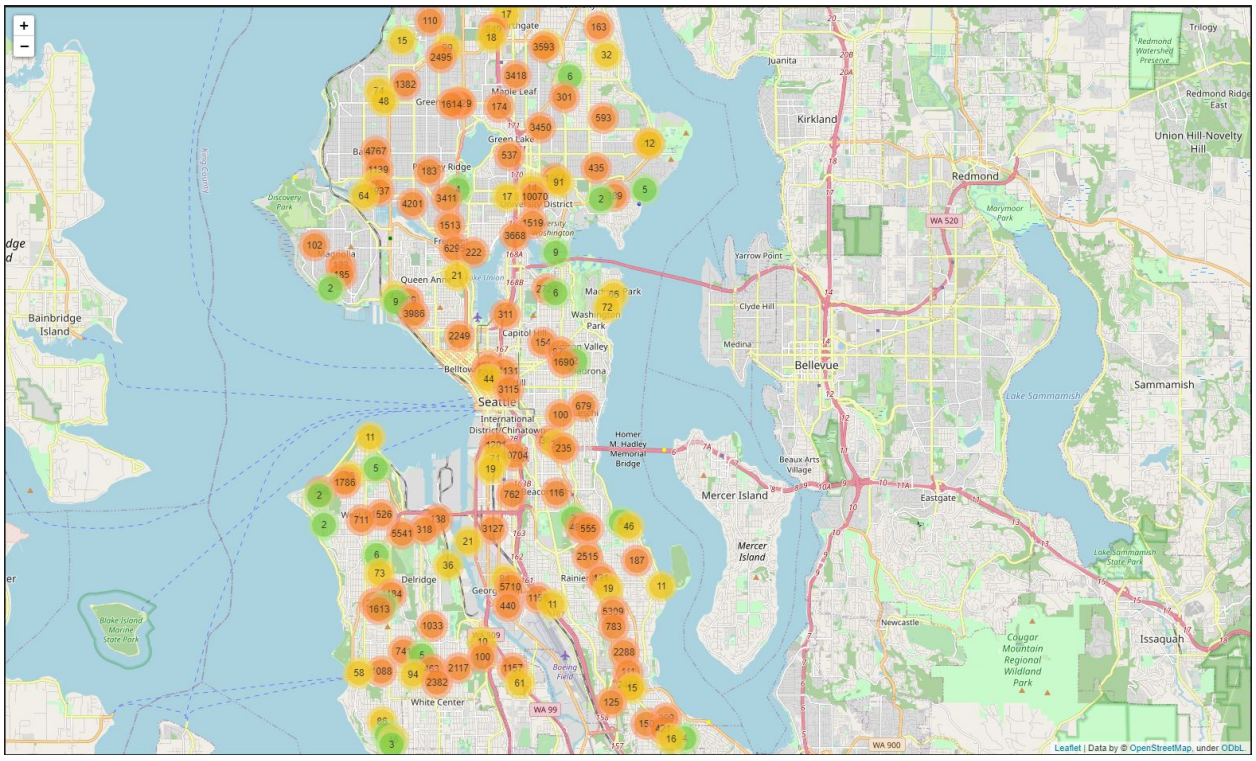
Different severity counts

So as we can see that the severity 1 and 2 are the most occurring ones. So we will focus on them only and filter out the rest.

And also do a random sampling to balance both datasets.

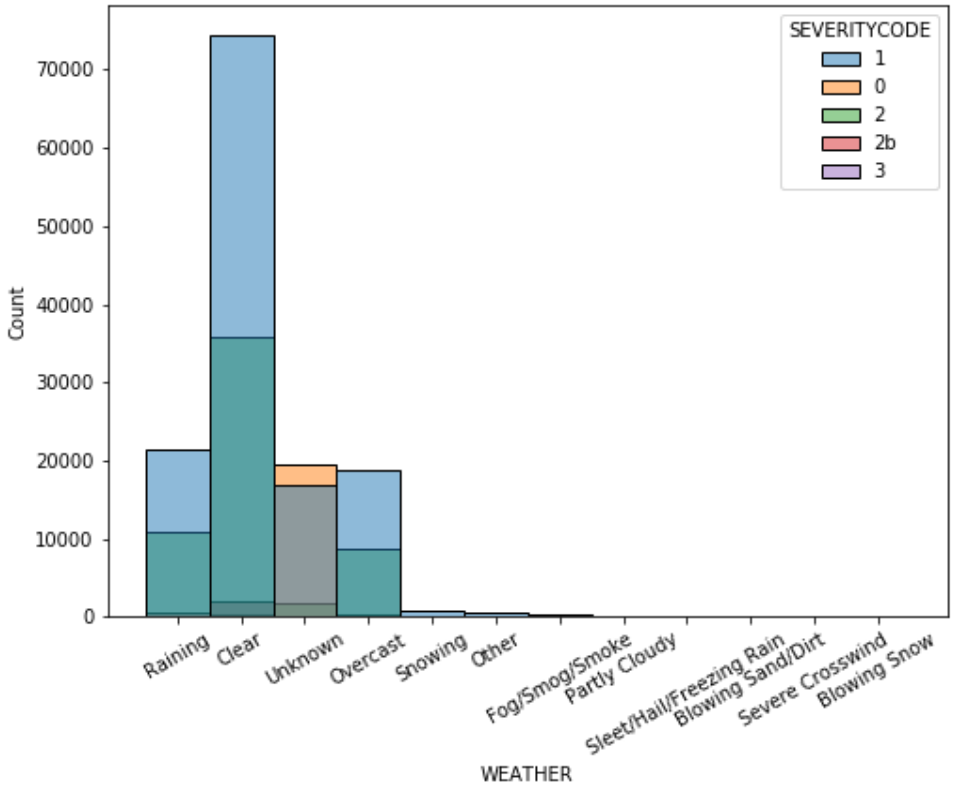
Data Analysis and Visualization

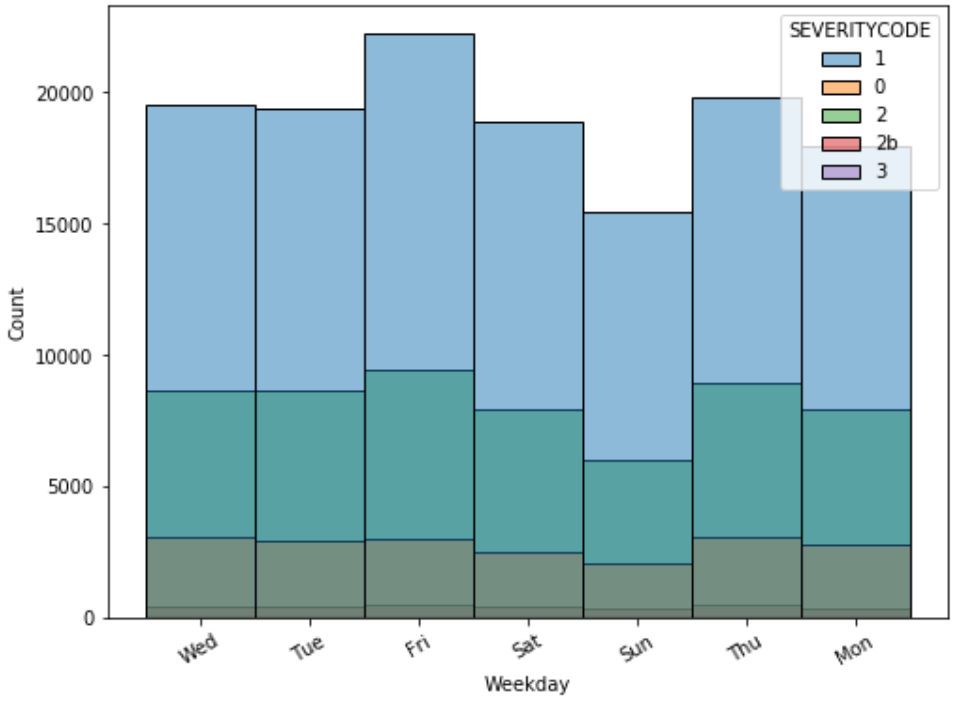
Seattle Accident Map

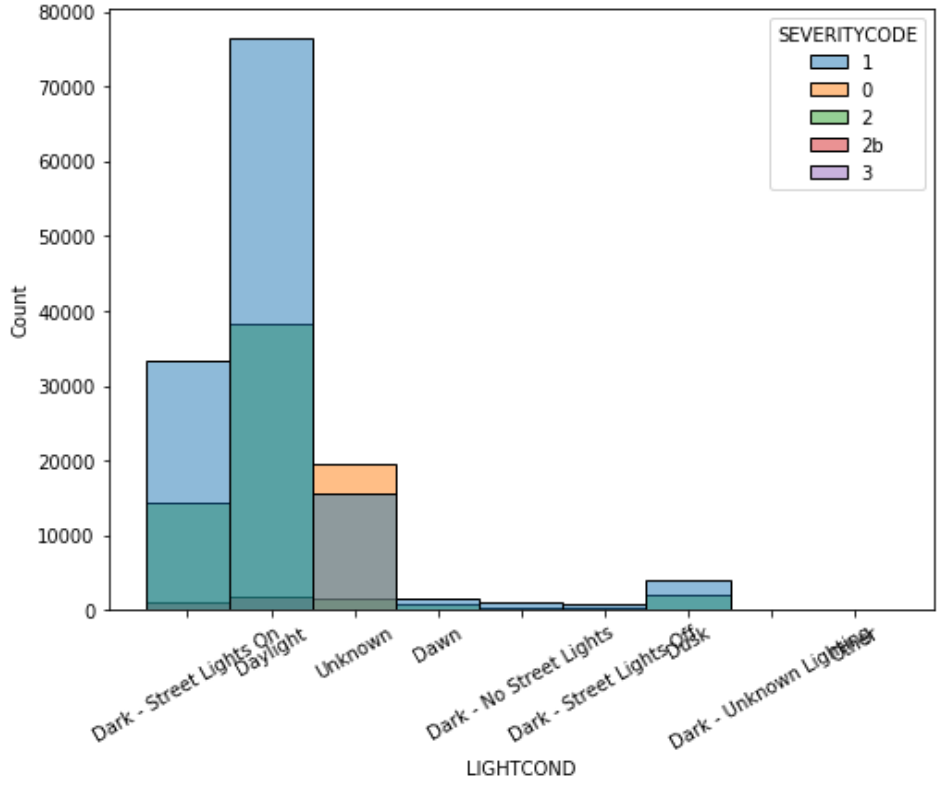


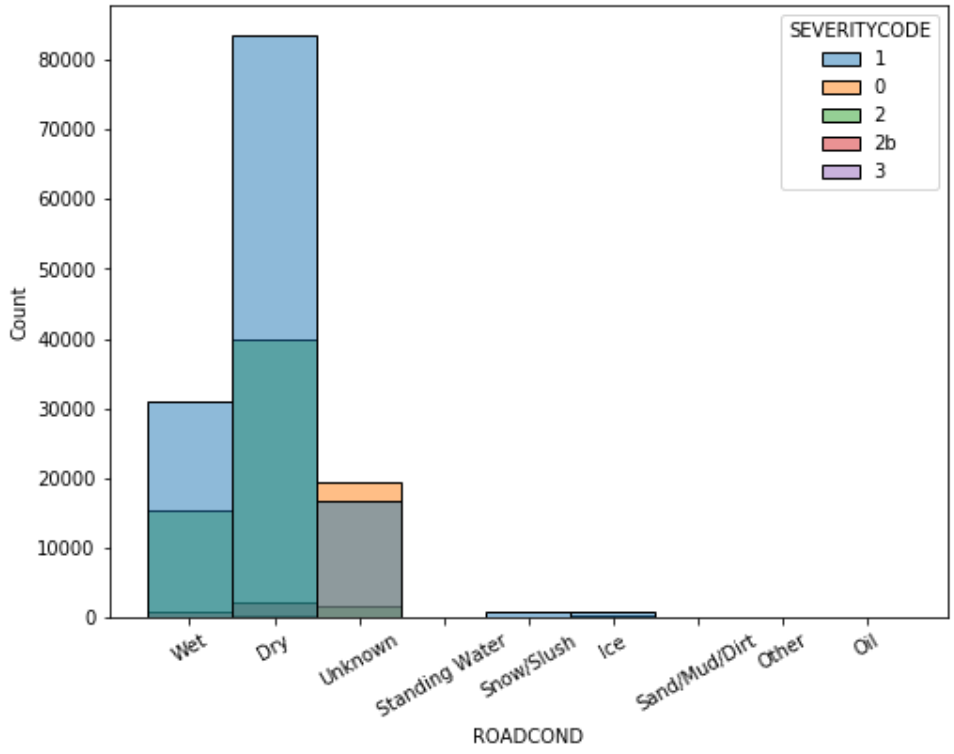
In this figure we can see the Seattle map and the accident clusters. The colours are dependent on the number of accidents.

Visualization based on various Columns


SEVERITYCODE
1
0
2
2b
3
70000
60000
50000
40000
30000
20000
10000
0
Count
Raining
Clear
Unknown
Overcast
Snowing
Other
Fog/Smog/Smoke
Partly Cloudy
Sleet/Hail/Freezing Rain
Blowing Sand/Dirt
Severe Crosswind
Blowing Snow
WEATHER


SEVERITYCODE
1
0
2
2b
3
20000
15000
10000
5000
0
Count
Wed
Tue
Fri
Sat
Sun
Thu
Mon
Weekday


SEVERITYCODE
1
0
2
2b
3
Count
80000
70000
60000
50000
40000
30000
20000
10000
0
Dark - Street Lights On
Daylight
Unknown
Dawn
Dark - No Street Lights
Dark - Street Lights Off
Dusk
Dark - Unknown Lighting
Other
LIGHTCOND


SEVERITYCODE
1
0
2
2b
3
Count
80000
70000
60000
50000
40000
30000
20000
10000
0
Wet
Dry
Unknown
Standing Water
Snow/Slush
Ice
Sand/Mud/Dirt
Other
Oil
ROADCOND

Deployment Modelling and Evaluation

For the Modelling, we will be using supervised learning, which means learning with class labels are already given in the dataset. Based on the combination of all independent features in the dataset, classification algorithms will predict the severity of accidents which is a binary classification.

Common classification algorithms which are used here is:

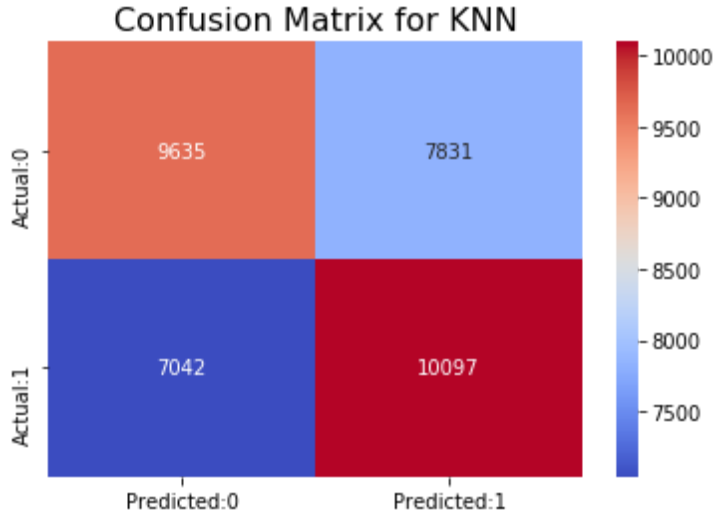
1. K-Nearest Neighbors
2. Decision Tree
3. Random Forest Classifier

For Modelling and Evaluation, the dataset will be split into Training and Testing subsets. The classification algorithm is trained to find the pattern that predicts the classes from the training subset, whereas the testing subset performs the accuracy testing. We will use several classification algorithms for modelling and testing to determine the appropriate model for predicting Accidents severity.

K-Nearest Neighbors (KNN)

It is a simple classification algorithm that uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. It works best when the dataset is balanced and its features are normalized. So we are building an accurate KNN model that determines the value of K, neighbours of comparison.

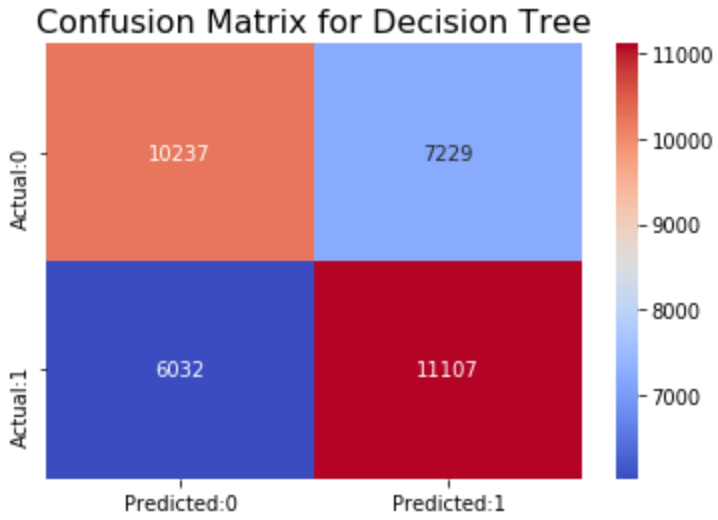
We found that the best k is 5.



Decision Tree

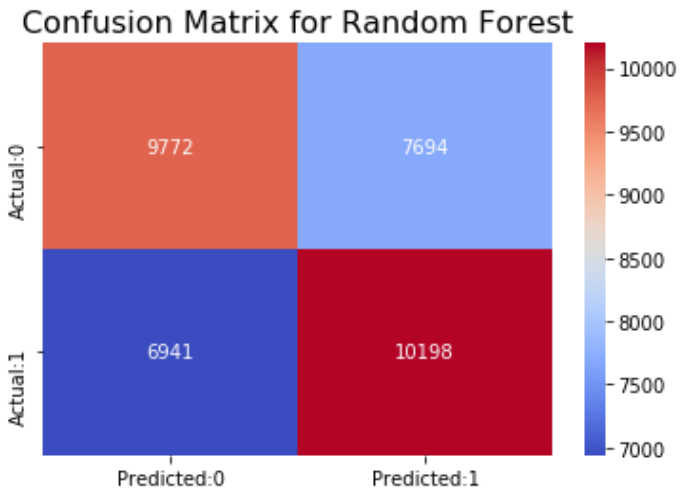
Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decision trees are the most powerful algorithms that fall under the category of supervised algorithms. In Decision Trees, leaves represent the class labels and branches represent conjunctions of features. Entropy defines the amount of information disorder, if the node is completely homogeneous (i.e.Single class), then the entropy is 0. If it is heterogenous then the entropy is 1.

We found that the best depth is 6.



Random Forest

Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.



Results

Based on the various results from different algorithms used as listed below, we were able to achieve an F1-score of 62 percent which is not very reliable in real life scenarios. But we can make further improvements using more feature engineering.

Algorithm	Jaccard	F1-score
KNN	0.57	0.57
Decision Tree	0.62	0.62
Random Forest	0.58	0.58

Discussion

Our Analysis mainly demonstrates the highest F1-score is achieved using Decision Tree classifier. This also suggests that we can use some better feature engineering to improve upon our results.

Conclusion

We can conclude that we can get an accuracy of more than 60 percent using a Decision tree model and can further improve upon the results using better features. So some future scopes are:

- Inclusion of as many features as possible by making the newly created instance from already existing one feature for better prediction
- Weighted XGBoost, Naïve Bayes and other similar models can be implemented instead of resampling the dataset.
- Detailed study of each feature and their correlation with dependent variable Severity will be done.